

## SPECIFICATION

## VARIANCE REDUCTION TECHNIQUE FOR LARGE SCALE RISK MANAGEMENT

Field of the Invention

This invention relates to methods for accelerating the convergence of Monte Carlo simulation for pricing portfolios of loans, mortgages or related financial instruments.

Description of Related Art

Financial institutions may carry large portfolios of mortgages and loans, numbering even in the millions of financial instruments. To evaluate the value of such portfolios, a number of Monte Carlo simulations are commercially available such as in the Oracle OFSA/Risk Manager software package. This particular software package is currently used by many large banks for evaluating their portfolios of mortgages, loans and deposits in what is called *Asset and Liability Management (ALM)*. Speed is a key concern for ALM packages for a number of reasons.

For example, estimating the value of such a portfolio using a Monte Carlo simulation with a single representative instrument is very inaccurate; therefore all instruments (which may number in the millions) are often processed individually by the simulation before accumulating the results. In addition, prepayment assumptions can be specified in a very sophisticated manner (at least in the OFSA/Risk Manager), which slows down the computations. Because of these factors, prior art Monte Carlo simulations tended to minimize the number of scenarios used while maintaining a satisfactory degree of accuracy.

A particularly difficult financial problem involves the evaluation of a portfolio of fixed rate mortgages, which are perhaps the most complicated of all banking instruments because of the path-dependency of amortization and prepayment. A simple Monte Carlo simulation of such a portfolio may require many thousands of samples before the simulation converges on a result with a suitable degree of accuracy. A number of variance reduction techniques may be used to reduce the required number of samples/scenarios used in a Monte Carlo simulation, including numerical integration using low discrepancy sequences (LDS) or control variates. The latter technique has not proven as popular as LDS or antithetic variables for several reasons. For instance, these techniques require an estimation phase to extract information about the system, adding overhead to the calculations. If only a few mortgages are in the portfolio, this extra overhead can be comparable to the time required to process the system with a simple Monte Carlo simulation.

These considerations do not apply, however, in evaluating a large portfolio. The computational overhead of a control variate technique is dwarfed by the sheer time of calculating the cash flows of a large portfolio. Also mortgages do not change very much with time. This means practically that one could estimate today's portfolio based on information from yesterday's portfolio, easing the estimation task of a control variate technique. Thus, there is a need in the art for improved control variate techniques that reduce the amount of scenarios required for a Monte Carlo simulation to converge to an answer or desired result.

### Summary of the Invention

The invention provides in one aspect a control variate technique for Monte Carlo simulation of the value of a portfolio of financial instruments. The control variate is formed by a

linear combination of an estimated value for the portfolio, an estimated aggregate value the portfolio, and an expected value of the estimated aggregate value. To derive the approximate simulated portfolio, a number of Monte Carlo samples are used to interpolate functions representing the approximate principal and approximate cashflow for the portfolio.

Other and further advantages of the invention will be further understood and appreciated by those skilled in the art by reference to the following specification, claims and drawings.

### Brief Description of the Drawings

The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate preferred embodiments of the invention, and together with the description, serve to explain the principles of the invention. Similar reference characters denote corresponding features consistently throughout the drawings.

Figure 1 is a flowchart of a simple Monte Carlo simulation.

Figure 2 is a flowchart of a control variate technique according to one embodiment of the invention.

Figure 3 is a graph comparing data obtained from a Monte Carlo simulation using a control variate technique according to one embodiment of the invention with data obtained from prior art Monte Carlo simulations.

Fig. 4 is a diagram of a computer hardware system with which the present invention can be implemented.

Fig. 5 is an additional diagram of computer hardware system with which the present invention can be implemented

## Detailed Description

The control variate technique of the invention may be used to calculate the *market value*, or *price* (at time zero) of a portfolio of mortgages. By definition the market value  $V_0$  of the portfolio is the sum of the market values  $V_0^j$  of each mortgage. As will be shown herein, this market value may be represented by an expected value of a stochastic integral. The integrand will depend upon the (risk-free) interest rate as well as the financial characteristics of the particular mortgage such as its coupon rate and the starting principal. To estimate a value for the integral, the integrand is sampled over the payment times of the mortgage. Various stochastic models or term structures may be used to generate the interest rates at these payment times. In a simple Monte Carlo simulation, values or samples of the integrand are calculated using an interest rate scenario produced by sampling a stochastic interest rate model with a pseudo-random number produced by a pseudo-random number generator. Multiple scenarios are produced and averaged to estimate the market value of the portfolio.

A popular stochastic model or term structure for the risk-free interest rate  $r$  is the Hull and White one-factor model. The stochastic differential equation (SDE) for the Hull and White one-factor model is given by:

$$dr_t = (\theta_t - ar_t)dt + \sigma dW_t \quad (1)$$

where  $a, \sigma$  are constants,  $W$  is one-dimensional Brownian motion in the risk-neutral measure, and  $\theta$  is a deterministic function of time that optimally fits the current yield curve. This means that given  $K$  observable discount bond prices  $P(T_k)$  for  $k = 1, \dots, K$ , a perfect choice of  $\theta$  would result in:

$$P(T_k) = E[e^{-\int_0^{T_k} r_s ds}] \quad k = 1, \dots, K \quad (2)$$

where  $E$  denotes time-zero expectation in the risk-neutral measure. By standard no-arbitrage arguments, the price of each mortgage then satisfies:

$$V_0^j = E\left[\sum_{i=1}^n e^{-\int_0^{T_i^j} r_s ds} C^j(T_i^j)\right] \quad (3)$$

where  $T_i^j$  are *payment times*, the *cash flows*  $C^j(T_i^j)$  are  $F_{T_i^j}$  measurable random variables that depend on mortgage characteristics, and  $n$  is the number of payments times required to retire the mortgage. Consider the case of fixed rate mortgages with prepayment optionalities. For convenience of notation but without loss of generality, it may be assumed that the maturity of each mortgage is  $T = 30$  years, and that the initial principal of each mortgage of the pool is the same. However, a typical mortgage portfolio consists of mortgages issued at different dates in the past, resulting in distinct values for the outstanding principal at time zero  $\pi^j$  of each mortgage. Moreover, the coupon rate  $\rho^j$  is different for each mortgage.

A Monte Carlo simulation is currently the most popular technique to calculate a value for equation (3). In that methodology, practitioners need to empirically estimate prepayment behavior. The most general endogenous model of prepayment is that of a functional of the rate, i.e. the time- $t$  prepayment rate is a functional  $\gamma_t : C([0, t], \mathfrak{R}) \otimes \mathfrak{R}_+^2 \rightarrow \mathfrak{R}_+$ , where  $C([0, t], \mathfrak{R})$  is the space of  $\mathfrak{R}$ -valued continuous functions of  $t$ . This prepayment model captures *path-dependent* behavior such as burnout and index lags. An example prepayment model is described further herein. The cash flow rate per unit of time  $\bar{C}^j$  and principal  $p$  follow then as:

$$\bar{C}_t^j = \begin{cases} \frac{\rho^j e^{\rho^j T}}{e^{\rho^j T} - 1} + \gamma_t(r, \rho^j, \pi^j) p_t^j & \text{if } p_t^j \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$\frac{dp_t^j}{dt} = -\frac{\rho^j e^{\rho^j \tau}}{e^{\rho^j \tau} - 1} + \rho^j p_t^j - \gamma_t(r, \rho^j, \pi^j) p_t^j \quad (5)$$

with  $p_0^j = \pi^j$ .

In a Monte Carlo simulation, the integral of equation (3) is sampled at discrete points to estimate its value. Thus, equations (1) and (4) must be discretized also. The Euler scheme may be used to discretize equation (1). Calculating a discrete cash flow from equation (4) is described by F. Fabozzi and F. Modigliani, *Mortgage and Mortgage-backed Securities Markets*, Harvard Business School Press, 1992. The *total cash flow rate*  $C$  is the sum of each mortgage cash flow rate and the *total principal*  $p$  is the sum of each mortgage principal.

A flowchart for a simple, prior art Monte Carlo simulation of equation (3) is illustrated in Figure 1. After starting the simulation, a random number generator generates a pseudo-random number to initiate the calculation of an interest rate scenario at step 105. For example, if the simulation is being implemented in C++, the function `random_fraction()` will provide a random number in the closed interval  $[0, \dots, 1]$ . Because a computer cannot generate a truly random number, the number is merely pseudo-random. With the appropriate weighting to maintain the required probability density, the pseudo-random number is plugged into the stochastic interest rate model at to generate interest rate samples at the payment times of the mortgage being evaluated. The stochastic interest rate model or term structure may be a discretized version of equation (3). Alternatively, other interest rate models may be used such as the Rendleman and Bartter model, the Vasicek model, or Hull and White two-factor model. Rather than use a pseudo-random number, LDS values could also be used in the stochastic interest rate model in step 105 to generate the interest rate samples such as described in U.S. Pat. No. 5,940,810, the contents of which are hereby incorporated by reference as if fully set forth herein. At step 110, the interest rate samples and financial characteristics of the mortgage, such as its starting

principal and coupon rate, are used to generate cash flow values at the payment times. These values are used to calculate samples of the integrand in equation (3) to estimate, for one scenario in the Monte Carlo simulation, the value for a particular mortgage at step 115. Steps 110 and 115 are repeated at step 120 until a value has been estimated for all the financial instruments in the portfolio.

The sum of all the values of all the mortgages calculated in step 120 provides an estimate of the value of the portfolio for one particular scenario in the Monte Carlo simulation. Many such scenarios must be repeated at step 125 depending upon the variance of the results. In a large portfolio, many different scenarios must be calculated before the simulation converges on a suitably accurate result. Because the process illustrated in Figure 1 implements no variance reduction techniques, it may be denoted as a simple or crude Monte Carlo simulation.

A control variate technique reduces the number of scenarios required for a Monte Carlo simulation to converge upon a result. The Monte Carlo simulation discussed with respect to Figure 1 may be generalized as providing an estimate of a function  $E(g(\mathbf{X}))$ , where  $\mathbf{X}$  is a random vector representing the interest rate samples at the payment times (a single Monte Carlo scenario). If another function  $f$  is derived whose expected value of  $f(\mathbf{X})$  is known, the convergence of the Monte Carlo simulation may be accelerated (assuming appropriate selection of a suitable  $f(\mathbf{X})$ ) by calculating samples of a new function  $W$ :

$$W = (g(\mathbf{X}) - f(\mathbf{X}) + m) \quad (6)$$

where  $m$  is the expected value of  $f(\mathbf{X})$ . The function  $W$  may be denoted as the control variate. The inventor has discovered a control variate that converges to a result more than two orders of magnitude faster than a simple Monte Carlo simulation.

In the control variate technique of the invention, the function  $f(X)$  represents the value of an aggregated approximate portfolio. Note that it is much less time-consuming to simulate the approximate portfolio than the true portfolio because the former is aggregated. As used herein, when references are made to an “aggregated” value such as an “aggregated cash flow value,” it will be understood that this refers to a value of the approximate portfolio, which does not simulate or model the financial instruments in the portfolio individually. To simulate the approximate portfolio, a functional form for its cash flow  $\hat{C}_i$  and principal  $\hat{p}_i$  for each month  $i$  (payment time) may be interpolated. The inventor has discovered that using two state variables, the interest rate and the approximate principal itself, provides a powerful control variate technique. The model of the approximate portfolio may be given by:

$$\hat{C}_i = \mu(r(i\Delta t), \hat{p}_i, i\Delta t) \quad (7)$$

$$\hat{p}_{i+1} = \nu(r(i\Delta t), \hat{p}_i, i\Delta t) \quad (8)$$

The functions  $\mu$  and  $\nu$  are determined by the information extracted in a first set of Monte Carlo scenarios, e.g.,  $\omega = 1, \dots, 50$ , using steps 105 through 125 in Figure 1. Because these scenarios are used to model an aggregate portfolio, they may be denoted as “adaptive” scenarios. Note that the value of 50 scenarios is arbitrary. The number of scenarios required to approximate the functions  $\mu$  and  $\nu$  is substantially less, however, than the many thousands of scenarios required in a simple Monte Carlo simulation.

From equation (3), the control variate  $W$  is thus given by:

$$W = \sum_{i=1}^n e^{-\int_0^{T_i^j} r_s ds} C^j(T_i^j) - \sum_{i=1}^n e^{-\int_0^{i\Delta t} r_s ds} \hat{C}_i(\omega) - \sum_{i=1}^n E[e^{-\int_0^{i\Delta t} r_s ds} \hat{C}_i] \quad (9)$$

where  $\hat{C}_i(\omega)$  is given by equation (7) and represents the cash flow for the approximate portfolio in the  $i$ th month (or  $i$ th payment time). An examination of equation (7) shows the dependence of



$\hat{C}_i(\omega)$  on the two state variables in the function  $\mu$ : the interest rate for the  $i$ th month,  $r(i\Delta t, \omega)$ , and the principal for the approximate portfolio at the  $i$ th month,  $\hat{p}_i(\omega)$ . Interpolating this function directly from the two state variables requires an interpolation in 3 dimensions. While such an interpolation can be done, there is a danger of overfitting the data in such a case.

Rather than interpolating in three dimensions, an interpolation in just two dimensions can be accomplished in the following manner. To decouple the dependence on the two state variables, the aggregate portfolio's cashflow for the  $i$ th month from only the payments (denoted as  $\hat{C}_i^{pay}$ ) is distinguished from the cashflow resulting from prepayments, and the dependence of  $\hat{C}_i^{pay}$  upon the principal is interpolated. In addition, the average prepayment rate for the aggregate portfolio in the  $i$ th month (denoted as  $\hat{\gamma}_i$ ) is derived as a function of  $r(i\Delta t)$ . Finally, the principal for the  $(i+1)$  month resulting only from the average prepayments in the  $i$ th month (denoted as  $\hat{p}_i^{net}$ ) is distinguished from the change in principal resulting from payments in the  $i$ th month, and the dependence of  $\hat{p}_i^{net}$  upon the principal is interpolated.

The adaptive scenarios are run (steps 105 through 125 as shown in Figure 1) and the following statistics are extracted, for each time  $i\Delta t$  and scenario  $\omega$ :

$$\text{average prepayment} = \sum_j \frac{\int_{t=i\Delta t}^{(i+1)\Delta t} \gamma_t p_t^j 1[p_t^j \geq 0] dt}{\hat{p}_i} \quad (10)$$

$$\text{total payment} = \sum_j \int_{t=i\Delta t}^{(i+1)\Delta t} \frac{\rho^j e^{\rho^j T}}{e^{\rho^j T} - 1} 1[p_t^j \geq 0] dt \quad (11)$$

where 1 is the indicator function, and  $j$  is the index for the  $j$ th financial instrument. A functional form for  $\gamma_t$  as a function of  $r(i\Delta t, \omega)$  may be derived in the following fashion. After graphing the average prepayment in ascending order of  $r(i\Delta t, \omega)$ , a decreasing piecewise linear function

(with, e.g., 5 evenly spaced anchor points) is interpolated from the data to represent  $\hat{\gamma}_i(r)$ , where  $\hat{\gamma}_i(r)$  represents the prepayment rate for the aggregate portfolio. Similarly, the total payment is graphed in ascending order of  $\hat{p}_i(\omega)$ , to which an increasing piecewise linear function (with, e.g., five evenly spaced anchor points)  $\hat{C}_i^{pay}(\hat{p}_i)$  is fitted. Finally, the ratio of  $\hat{p}_{i+1}$  over (one minus the average prepayment for the  $i$ th month) is plotted in ascending order of  $\hat{p}_i$  to linearly interpolate the function  $\hat{p}_i^{net}$ . The function  $\mu$  and  $\nu$  are then defined by:

$$\mu(r, p, i\Delta t) = \hat{C}_i^{pay}(p) + p\hat{\gamma}_i(r) \quad (12)$$

$$\nu(r, p, i\Delta t) = \hat{p}_i^{net}(p)(1 - \hat{\gamma}_i(r)) \quad (13)$$

It will be understood that the use of 5 anchor points is arbitrary as well as using piecewise linear functions. Other functional forms could be used with varying number of anchor points to approximate the functions  $\hat{\gamma}_i(r)$ ,  $\hat{C}_i^{pay}$ , and  $\hat{p}_i^{net}$ .

An additional set of Monte Carlo scenarios may be used to form the control variate  $W$  of equation (9). A flowchart for the control variate technique of the invention is illustrated in Figure 2. Upon starting the simulation, the statistics necessary to derive the average prepayment and total payment (equations (10) and (11)) are generated in a set of adaptive Monte Carlo scenarios at step 200. These scenarios are generated in an analogous fashion to steps 105 through 125 described with respect to Figure 1. The average prepayment and total prepayment may then be calculated using discrete forms of equations (10) and (11) at step 205. At step 210, the functions  $\hat{\gamma}_i(r)$ ,  $\hat{C}_i^{pay}(\hat{p}_i)$ , and  $\hat{p}_i^{net}(\hat{p}_i)$  are interpolated.

To begin calculating the control variate  $W$  given by equation (9), the simulation begins another set of Monte Carlo scenarios at step 215 by generating an interest rate scenario at step 215. At step 220, the interest rate scenario is used to calculate a value for the first term in

equation (9). This gives a value of the simulated portfolio according to one Monte Carlo simulation as described in steps 110, 115, and 120 with respect to Figure 1. The value of the simulated aggregated portfolio may now be calculated based upon the interest rate scenario provided by step 215. Note that equation (10) predicts the value for the aggregate principal for the  $(ith + 1)$  month based upon the interest rate and the aggregate principal for the  $ith$  month. Thus, based upon the known starting principal values for the mortgages in the portfolio and the interest rate samples, the aggregate principal values may be sequentially derived using equation (12) at step 225. In turn, the interest rate scenario and the aggregate principal values are substituted into equation (11) to calculate the aggregate cash flow values at step 230. These values are used in a discretized approximation of the second term in equation (9) to give a value of the simulated aggregate portfolio according to one Monte Carlo simulation at step 235. Not illustrated in the flowchart is a step to calculate the expected value of the simulated aggregate portfolio. As will be described further herein, this term may be calculated by a simple Monte Carlo simulation or by a lattice approximation. The resulting expected value and the values from steps 235 and 220 are linearly combined at step 240 to calculate the control variate. Additional Monte Carlo scenarios are repeated at step 245 until the simulation converges to a result at step 250.

The aggregate cash flow and principal given by equations (7) and (8) is a rough approximation of the values given by equation (4)-(5). For instance, it does not capture burnout or index lags. The inventor has discovered, however, that most of the path-dependency of residential mortgages comes from principal amortization. Thus, an inventive solution is to take the approximate principal as a second state variable. This is a relatively optimum selection of

state variables since taking more than two state variables complicates the estimation phase, requiring interpolation in more than two dimensions.

A simple Monte Carlo simulation may be used to calculate (with a very large number of scenarios) the expected value term for the control variate since the value of the aggregate portfolio is inexpensive to compute. However, computing the expected value with a lattice, i.e. a sparse representation of a Markov chain, provides a hard bound on the bias so that this technique is preferable. This is possible because the two state variables  $(r, \hat{p})$  constitute a joint Markov process.

The lattice approximation of the invention is three dimensional and is an extension of the two-dimensional lattice discussed in Hull, *Options, Futures, and Other Derivatives*, p.438, Prentice-Hall, 1997, 3<sup>rd</sup> Ed. to model instruments whose cash flows  $C$  depend on the interest rate only. The main features of the latter are now discussed to provide background for the novel lattice approximation of the invention. In Hull and White's two-dimensional lattice, time takes the discrete values  $i\Delta t$ , and the discretized interest rate process  $r_d(i)$  takes the discrete values  $\alpha_{i,j} \equiv E[r(i\Delta t)] + j\Delta r$ , where the rate-step  $\Delta r$  is appropriately chosen for error minimization. The transition probabilities at node  $(i, j)$  of the lattice are zero, except for 3 nodes  $(i+1, j_{next}-1)$ ,  $(i+1, j_{next})$ , and  $(i+1, j_{next}+1)$ ; they are such that the first two moments of the discretized process  $r_d$  for the rate equal the first two moments of the process  $r$  over a time-step  $\Delta t$ . To calculate the market value  $V$ , equation (3) is discretized:

$$V = \sum_{i=1}^n \sum_j Q(i, j) C(i\Delta t, j\Delta r) \quad (14)$$

where  $Q(i, j)$  is the price of an Arrow-Debreu security that pays 1 if node  $(i, j)$  is reached, i.e.:

$$Q(i, j) = E[e^{-\sum_{k=0}^{i-1} r_d(k)\Delta t} | r_d(i) = \alpha_{i,j}] \quad (15)$$

Note that this lattice has only two dimensions: time (through  $i$ ) and interest rate (through  $j$ ). In an embodiment of the lattice used to calculate the expected value term for the control variate, a dimension is added to account for dependence on the principal, such that the node  $(i, j, k)$  is a time  $i\Delta t$  state where the rate equals  $\alpha_{i,j}$  as before, and the principal equals  $k\Delta p$  for some value of the principal step  $\Delta p$ . Conditional on that information, the principal at time  $(i+1)\Delta t$  will be equal to  $v(\alpha_{i,j}, k\Delta p, i\Delta t)$  because of equation (8). At each node  $(i, j, k)$ ,  $k_{next}$  can be determined by rounding, so that:

$$v(\alpha_{i,j}, k\Delta p, i\Delta t) - dp / 2 \leq k_{next} dp \leq v(\alpha_{i,j}, k\Delta p, i\Delta t) + dp / 2 \quad (16)$$

Since the dynamics of the principal are deterministic between two time-steps, it is only possible to jump from node  $(i, j, k)$  to nodes  $(i+1, j_{next} + 1, k_{next})$ ,  $(i+1, j_{next}, k_{next})$ , or  $(i+1, j_{next} - 1, k_{next})$ . The transition probabilities are the same as in the two-dimensional lattice.

The market value of the approximate portfolio is then:

$$\sum_{i=1}^n \sum_j \sum_k Q(i, j, k) \mu(j\Delta r, k\Delta p, i\Delta t) \quad (17)$$

where  $Q(i, j, k)$  is the price of an Arrow-Debreu security that pays 1 if the node  $(i, j, k)$  is reached.

*Numerical results.* An embodiment of the control variate technique of the invention was tested

on a portfolio of 100 mortgages. The test used  $p_0^j = j$ ,  $\rho^j = 0.062 + 0.0002 j - 0.02 *$

$(j/100)^2$  for  $j = 1, \dots, 100$ . The rate spread  $x$  and the prepayment function  $\gamma$  are chosen to be:

$$x = \max[\rho^j - \frac{r(i\Delta t) + r((i-1)\Delta t)}{2} - 0.005, 0] \quad (18)$$

$$\gamma(i\Delta t) = \begin{cases} \min[0.01 + 10x + 5000x^3, 1] / \Delta t & \text{if } i \leq 4; \\ 0 & \text{otherwise.} \end{cases} \quad (19)$$

The term structure parameters are  $a = 0.1$ ,  $\sigma = 0.014$ , and the yields of the discount bonds are given in the following Table 1. Cubic splines are used to interpolate these yields in order to fit  $\theta$  according to equation (2).

**Table.1. Original Yield Curve**

Maturity	0.08	0.17	0.25	0.33	0.42	1
Yield (in %)	5	5.23	6	6.2	6.5	6.5

The results are shown in Figure 3. In this test, the interest rate samples were generated using LDS samples for input into a stochastic interest rate model rather than with a pseudo-random number generator. Figure 3 compares the absolute value of the relative error (compared to a simulation of 100,000 pseudo-random numbers) of three estimators of the price of the mortgage portfolio: LDS, LDS with control variates, and pseudo-random numbers. For the latter, the average absolute value of the integration error is shown over 10 different seeds. For control variates, the 50 scenarios from the adaptive scenarios were not shown in the results since in practice this data is readily available from previous simulations of the portfolio. As mentioned earlier, the overhead to generate the control variate is negligible. This experiment demonstrates that, for a mortgage portfolio, integration error is about 5 times lower with LDS than with pseudo-random numbers. If control variates with LDS is used, it is about 20 times lower. The variance of the pseudo-random estimator is between 100 and 300 times higher than the variance of the control variate estimator with LDS.

An example computer system for implementing an embodiment of the invention is illustrated in Figure 4. Referring to Fig. 4, in an embodiment, a computer system 720 includes a host computer 722 connected to a plurality of individual user stations 724. In an embodiment,

the user stations 724 each comprise suitable data terminals, for example, but not limited to, e.g., personal computers, portable laptop computers, or personal data assistants ("PDAs"), which can store and independently run one or more applications, i.e., programs. For purposes of illustration, some of the user stations 724 are connected to the host computer 722 via a local area network ("LAN") 726. Other user stations 724 are remotely connected to the host computer 722 via a public telephone switched network ("PSTN") 728 and/or a wireless network 730.

In an embodiment, the host computer 722 operates in conjunction with a data storage system 731, wherein the data storage system 731 contains a database 732 for storing financial information about a portfolio of financial instruments that is readily accessible by the host computer 722. Note that a multiple tier architecture can be employed to connect user stations 724 to a database 732, utilizing for example, a middle application tier (not shown).

In alternative embodiments, the database 732 may be resident on the host computer, stored, e.g., in the host computer's ROM, PROM, EPROM, or any other memory chip, and/or its hard disk. In yet alternative embodiments, the database 732 may be read by the host computer 722 from one or more floppy disks, flexible disks, magnetic tapes, any other magnetic medium, CD-ROMs, any other optical medium, punchcards, papertape, or any other physical medium with patterns of holes, or any other medium from which a computer can read. In an alternative embodiment, the host computer 722 can access two or more databases 732, stored in a variety of mediums, as previously discussed.

Referring to Fig. 5, in an embodiment, each user station 724 and the host computer 722, each referred to generally as a processing unit, embodies a general architecture 805. A processing unit includes a bus 806 or other communication mechanism for communicating instructions, messages and data, collectively, information, and one or more processors 807

coupled with the bus 806 for processing information. A processing unit also includes a main memory 808, such as a random access memory (RAM) or other dynamic storage device, coupled to the bus 806 for storing dynamic data and instructions to be executed by the processor(s) 807. The main memory 808 also may be used for storing temporary data, i.e., variables, or other intermediate information during execution of instructions by the processor(s) 807.

A processing unit may further include a read only memory (ROM) 809 or other static storage device coupled to the bus 806 for storing static data and instructions for the processor(s) 807. A storage device 810, such as a magnetic disk or optical disk, may also be provided and coupled to the bus 806 for storing data and instructions for the processor(s) 807.

A processing unit may be coupled via the bus 806 to a display device 811, such as, but not limited to, a cathode ray tube (CRT), for displaying information to a user. An input device 812, including alphanumeric and other keys, is coupled to the bus 806 for communicating information and command selections to the processor(s) 807. Another type of user input device may include a cursor control 813, such as, but not limited to, a mouse, a trackball, a fingerpad, or cursor direction keys, for communicating direction information and command selections to the processor(s) 807 and for controlling cursor movement on the display 811.

According to one embodiment of the invention, the individual processing units perform specific operations by their respective processor(s) 807 executing one or more sequences of one or more instructions contained in the main memory 808, wherein these sequences of instructions carry out the control variate technique of the invention. Such instructions may be read into the main memory 808 from another computer-usable medium, such as the ROM 809 or the storage device 810. Execution of the sequences of instructions contained in the main memory 808 causes the processor(s) 807 to perform the processes described herein. In alternative



embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and/or software.

The term “computer-usable medium,” as used herein, refers to any medium that provides information or is usable by the processor(s) 807. Such a medium may take many forms, including, but not limited to, non-volatile, volatile and transmission media. Non-volatile media, i.e., media that can retain information in the absence of power, includes the ROM 809. Volatile media, i.e., media that can not retain information in the absence of power, includes the main memory 808. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise the bus 806. Transmission media can also take the form of carrier waves; i.e., electromagnetic waves that can be modulated, as in frequency, amplitude or phase, to transmit information signals. Additionally, transmission media can take the form of acoustic or light waves, such as those generated during radio wave and infrared data communications.

Common forms of computer-usable media include, for example: a floppy disk, flexible disk, hard disk, magnetic tape, any other magnetic medium, CD-ROM, any other optical medium, punchcards, papertape, any other physical medium with patterns of holes, RAM, ROM, PROM (i.e., programmable read only memory), EPROM (i.e., erasable programmable read only memory), including FLASH-EPROM, any other memory chip or cartridge, carrier waves, or any other medium from which a processor 807 can retrieve information.

Various forms of computer-usable media may be involved in providing one or more sequences of one or more instructions to the processor(s) 807 for execution. For example, the instructions may initially be provided on a magnetic disk of a remote computer (not shown). The

remote computer may load the instructions into its dynamic memory and then transit them over a telephone line, using a modem. A modem local to the processing unit may receive the instructions on a telephone line and use an infrared transmitter to convert the instruction signals transmitted over the telephone line to corresponding infrared signals. An infrared detector (not shown) coupled to the bus 806 may receive the infrared signals and place the instructions therein on the bus 806. The bus 806 may carry the instructions to the main memory 808, from which the processor(s) 807 thereafter retrieves and executes the instructions. The instructions received by the main memory 808 may optionally be stored on the storage device 810, either before or after their execution by the processor(s) 807.

Each processing unit may also include a communication interface 814 coupled to the bus 806. The communication interface 814 provides two-way communication between the respective user stations 824 and the host computer 822. The communication interface 814 of a respective processing unit transmits and receives electrical, electromagnetic or optical signals that include data streams representing various types of information, including instructions, messages and data.

A communication link 815 links a respective user station 824 and a host computer 822. The communication link 815 may be a LAN 726, in which case the communication interface 814 may be a LAN card. Alternatively, the communication link 815 may be a PSTN 728, in which case the communication interface 814 may be an integrated services digital network (ISDN) card or a modem. Also, as a further alternative, the communication link 815 may be a wireless network 730.

A processing unit may transmit and receive messages, data, and instructions, including program, i.e., application, code, through its respective communication link 815 and

communication interface 814. Received program code may be executed by the respective processor(s) 807 as it is received, and/or stored in the storage device 810, or other associated non-volatile media, for later execution. In this manner, a processing unit may receive messages, data and/or program code in the form of a carrier wave.

Those of ordinary skill will appreciate that many different embodiments of the invention may be constructed. Thus, it is understood that other embodiments of the present invention can be fabricated and be within the spirit and scope of the appended claims.

[illegible]